

Python Smart ClipBoard

 opencircuits.com/Python_Smart_ClipBoard

Goal/Purpose

This Python program (clip_board.py) is an application to enhance the windows (unix ??) clipboard by adding the ability to it to:

- Transform keyboard contents, referred to as "Transforms".
- Insert predefined elements into the clipboard, referred to as "Snippets".
- Open example files in a configurable editor or ide, referred to as "Snips". -- This really does not involve the clipboard, but is a sort of high powered version of snippets
- Interpret copied text as commands or directions to the system to carry out some action specified in the clipboard contents, for example, browse to some website.
- The program is easily modified by adding transforms, snips, or snippets, this may require some python programming.

The program does not have a clipboard history as I, so far, have not found that useful. There are a lot of other apps with this feature.

See the GUI at [ClipBoard GUI](#)

Status

This program is up and running most of the functions seem to work well, major problem is that I keep adding details to the functionality, I use the program partly as an opportunity to explore Python. Making improvements from time to time for my projects. Code at GitHub, see [\[Code at GitHub\]](#) I think the major problem with the code is that I am still making a fairly large number of changes/enhancements. As a result I am breaking things.

Code

Code at GitHub, see [\[Code at GitHub\]](#)

This is an article started by Russ Hensel, see "[http://www.opencircuits.com/index.php?title=Russ_hensel#About My Articles](http://www.opencircuits.com/index.php?title=Russ_hensel#About_My_Articles)" About My Articles for a bit of info.

Who should use this program and How:

Who	What	How
Person with little programming experience, no interest in Python. Looking for download and install.	Probably should use another program.	Not well suited to use this, I do not plan to fix this.
Likely a programmer or other user who frequently needs to transform text or insert standard snippets in text.	Run the basic program. Customize in fairly minor ways.	Edit the parameter file to configure the program.
Modest Python experience, same needs as above.	Modify all over the place, add/delete functionality.	Program should be well documented with an API, sample extensions. Code changes localized (now) to one class plus the parameter file.

What/Why

As a programmer and general computer hacker I often need to make simple but time consuming changes to text or recall easily forgotten text, a couple of examples:

In a wiki you can hide text with a bit of special markup: it uses < and > and --- and ! I often forget exactly

what it looks like, and in any case it is a pain to type. If I click on "wiki hidden" in the clipboard app then the text is put in the clipboard and I can paste in the wiki.

```
<!-------
this cannot bee seen
----->
```

In a similar way a wiki has a big messy way of setting up a table. "wiki table" puts a whole example in my clipboard for pasting.

Many programmers like file names without spaces. A transformation function in the clipboard will take the text in the clipboard, remove the spaces and put it back in the clipboard.

Some tasks like opening text files or url can be made easy from text you store, for this a prefix is used. To open a text file copy from a place you have stored it:

```
*>text some_file.txt
```

Then pressing the <*cmd> button will open the file in your chosen text editor. In a similar way;

```
*>url http://www.opencircuits.com/Main_Page
```

Will open the main page of Open Circuits in your chosen browser.

What are some other features?

- Free open source
- Runs across OSs Linux (inc **Raspberry Pi**), Mac or Windows
- Python
- Parameter file for wide range of modifications of program behavior.
- Uses standard Python logging class.

What would you like to see in the program or documentation? Email me.

Installation

This program is intended for those who at least occasionally develop in Python. I expect that they already run some things in Python and will just add this as an additional project. There is no install program you just download the files, place where you keep your projects and run. A bit more later in this section.

My Environment/Your Environment

The program has a better chance of running if your environment is not too much different from mine. The most important is that it is Python 3.6 or higher

Before you begin to install you should know a bit about the environment that I have used to build, test and run the terminal. If your environment differs too much you may have trouble getting it to run.

I run Python mostly using the install that comes with Anaconda Spyder and often use the IDE it installs. This is not necessary, it is just a nice install that downloads a lot of stuff that technical folks find useful. I have used conda and pip to add to this install and do not know offhand all that is in it. You can look at the include statements to get some idea of what you might need to add. Or you can just keep running it and add the packages it complains about.

Download

Code at GitHub, see [\[Code at GitHub\]](#) (it is Python and you can run directly from the source) Email me if you have issues (use this link [User:Russ_hensel](#)). You will get a zip file, unzip it and you should get:

```
.... whatever --|
                |
                |-- clipboard  --| -> all code required to run the application ( not sure if smart_terminal or
python_smart_terminal or nothing is top level name, just put it in some well named place )

                | -- images -> image files, mostly screen shots, icons... or what ever, not
important for the code.

                | -- wiki_etc -> various files documenting program, including at least some of
the material from this wiki
```

Put them in your system making "....whatever" anything convenient for your Python (that is move the files to where you keep your Python source).

Note that there may be a certain amount of left over, dead code, in the directory I am cleaning out bit by bit, someday it may be nice and neat. For now if you want to tinker look at the design info below first.

Run

Run it until it stops complaining about dependencies (in the console), after that (and perhaps even before) the GUI should come up. You are installed.

I have run the program on both Windows 10 but not Linux or Rasperian on a RPi. It should may in most OS's, this depends on how well pyclip a library I use works on these OS. Let me know about issues.

Configure to Run

Basic

Basic configuration is all done in a file called parameters.py. It seemed easier to simply use a Python text file instead of some other format like an ini file. Pretty much all the file does is set instance variables in itself. It is used by the program controller (clip_board.App) to create an instance of Parameters and then the values can be used. Save the original (parameters.py, maybe I will include a backup maybe not) in case you mess it up too much.

I have made yet another pass to clean up and comment the code in parameters. Let me know if you have issues. You should understand some values are being phased out but may still have some implementation and some may be coming in and have little or no implementation. The comments should let you identify these situations.

Parameters starts out with some "meta" parameters. These are defined early in the creation of the objects and may effect other values. In any case you can always define a value twice, the last one always wins. The most important meta parameter is mode, you should not change it from self.mode = "Terminal" unless you understand the implications or do not mind going on a ride.

All the com port values are defined in pretty much one place, find it (say search on "baud") an change it to what you need.

Clipboard Configuration config.....

Running It

When you run it it should open a windows a lot like the picture [[ClipBoard GUI]. Errors may show up in your Python console or the log file (look in parameters.py for the name of the log file, typically self.pylogging_fn = "smart_terminal.py_log"). The most likely errors will point to missing Python modules like pyperclip. You should install with pip (or conda if using Spyder). Let me know how it goes.

Setting up for your system editor:

First configure parameters.py to know the name of a text editor on your system. For mine this is one of:

```
self.ex_editor = r"leafpad" # linux and pi
```

```
self.ex_editor = r"D:\apps\Notepad++\notepad++.exe" # for windows.
```

It is set up to auto switch between the two os to make copying the whole program back and forth between the windows and linux a bit easier.

Now when you run it the button <Edit Parms> should let you edit the parameters.py file. Edit it and save.

Hit the <Restart> button. In a flash the program should restart with the new parameters.

..... more here soon

Notes on the Code

Code quality is uneven some dates back a long time ago, some has ideas I tried, but now do not like. There is too much dead code, ass backwards was of doing things, poor naming..... There is also a good bit of code on features that I am adding, but that is unfinished. Until I loose interest in it it will probably improve. However, adding features is more part of the life of a programmer than polishing old features; making something better that seems good enough is not always the priority it should be. If you do not like it, mostly keep it to yourself unless it is accompanied by an offer to improve it. I do not need ideas, I need time. That said if you think you have a helpful comment contact me, my page will tell you how:

User:Russ_hensel

Design

Description of the design, and a bit to help you figure out what the files do (for .py files also see the top of file).

see:

Customizing/Extending

Simple customization may be done simply by changing the parameter file, see: [Smart Terminal Parameter Examples](#) If you want to add code that can be done by messing with any of the source code (not recommended) or by programming an extension, see: [Python Smart Terminal Technical](#) and [Writing You Own Extensions to SmartTerminal](#)

Additional Info

Click on the category **ClipBoard** below (and perhaps the others as well)

[Writing You Own Extensions to ClipBoard](#)

[ClipBoard at git hub](#)